CONTENTS

NASA Technical Paper 1817

Time and Frequency Domain Analysis of Sampled Data Controllers via Mixed Operation Equations

Harold P. Frisch

JUNE 1981

NASA

NASA Technical Paper 1817

Time and Frequency Domain Analysis of Sampled Data Controllers via Mixed Operation Equations

Harold P. Frisch
Goddard Space Flight Center
Greenbelt, Maryland

NASA
National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

1981

All measurement values are expressed in the International System of Units (SI) in accordance with NASA Policy Directive 2220.4, paragraph 4.

ABSTRACT

Specification of the mathematical equations required to define the dynamic response of a linear continuous plant, subject to sampled data control, is complicated by the fact that the digital components of the control system cannot be modeled via linear ordinary differential equations. This complication can be overcome by introducing two new mathematical operations; namely, the operation of zero order hold and digital delay. It is shown herein that by direct utilization of these operations, a set of linear mixed operation equations can be written and used to define the dynamic response characteristics of the controlled system. It also is shown how these linear mixed operation equations lead, in an automatable manner, directly to a set of finite difference equations which are in a format compatible with follow-on time and frequency domain analysis methods.

Blank Page

CONTENTS

TIME AND FREQUENCY DOMAIN ANALYSIS OF SAMPLED DATA CONTROLLERS VIA MIXED OPERATION EQUATIONS

Harold P. Frisch
Goddard Space Flight Center
Greenbelt, Maryland

INTRODUCTION

The large space systems technology (LSST) program of the National Aeronautics and Space Administration (NASA) is attempting to develop an interactive integrated analysis capability (IAC). Its objective is to provide engineers with an interdisciplinary (structures-thermal-controls) analysis capability supportive of static, time and frequency domain design and performance evaluation methods. To do this, the IAC program intends to provide a time-wise efficient and user-friendly means for accessing a sequence of analysis tools and for respectively obtaining the necessary input data via a series of established interdisciplinary data flow paths. The immediate goal is to provide a coupled structures-thermal-controls analysis capability compatible with the inherent assumptions and limitations associated with current widely used state-of-the-art general purpose analysis programs such as NASTRAN, DISCOS, SINDA and TRASYS.*

This stated goal implies that data must be derived and stored in computer memory in such a manner that it can flow between analysis modules which view intrinsically identical items of data differently. Today, the prime impediment to this interdisciplinary flow of data is the infinite number of ways that requested data can be generated and provided to the requestor. In a working environment, this problem is usually compounded by the inability of data requestors to cross interdisciplinary semantics barriers and to define precisely what is being requested. The problem is particularly acute when data must be obtained from sources outside of one's immediate working environment. The net effect of these data flow barriers is a drop in analyst productivity while assumptions and limitations associated with data delivered in an unfamiliar data format are investigated.

One technique which can be used to overcome this real world problem is to define neutral formats for data. Conceptually, once data is in a neutral format it can, with the aid of a processor, flow anywhere. Since both the neutral format and the desired format are known a priori, the development of the processor which creates the linkage becomes a routine task. In order to minimize the number of processors required, the choice of a neutral format converges on the answer to the question: What is the most commonly used, readily obtainable format for requested data? For example, in the structural analysis discipline NASTRAN is probably the most widely used general purpose analysis program. Its bulk data deck would be a good choice as the neutral format for structures data.

The choice of a neutral format for thermal analysis is complicated by the fact that there are two competing methods of analysis. Thermal engineers having a strong structures background usually lean toward the finite element approach and use either the NASTRAN or SPAR thermal analyzers. Thermal

*See Program Acronym/Availability List, page 18.

engineers having limited or no structures background usually lean toward the finite difference approach and use the programs SINDA and TRASYS. Consequently, a bilevel neutral format will most probably be required for this discipline. Level one would contain the finite element neutral format while level two would contain the finite difference neutral format. If NASTRAN and SINDA bulk data decks are selected as neutral formats, then it is possible to generate an automated procedure which will read level 1 NASTRAN bulk data and generate from it level 2 SINDA bulk data. The reverse probably never will be possible.

The choice of a single or multilevel neutral format for controls data is not at all straightforward. If the assumption could be made, which it cannot, that controllers are continuous system devices, then the input data for any of the continuous system modeling programs such as ASCL, CSMP, CSSL, DARE, MODEL, etc., would be candidates for a neutral format for controls data. Since most present-day spacecraft controllers are sampled data rather than continuous devices, new modeling programs must be developed or old ones extended. This is required to adequately account for, in a simultaneous manner, the continuous time response characteristics of the plant and the digital response characteristics of controller components.

The objective of this paper is to present a framework for a neutral format which can be used for specification of a full range of sampled data controllers. It is conceptually simple in that it parallels the block diagram approach conventionally used for control system documentation. Furthermore it can be computationally automated in such a manner that a clear distinction is made between state variables which may be labeled as continuous time or analog signals and those which may be labeled as digital signals. This clear distinction allows for the development of integration algorithms which utilize the discontinuous nature of digital signal state variables to improve run-time performance. When the system is linear, this framework also leads directly to linear constant coefficient finite difference equations compatible with stability and frequency domain analysis methods.

MIXED OPERATION EQUATIONS

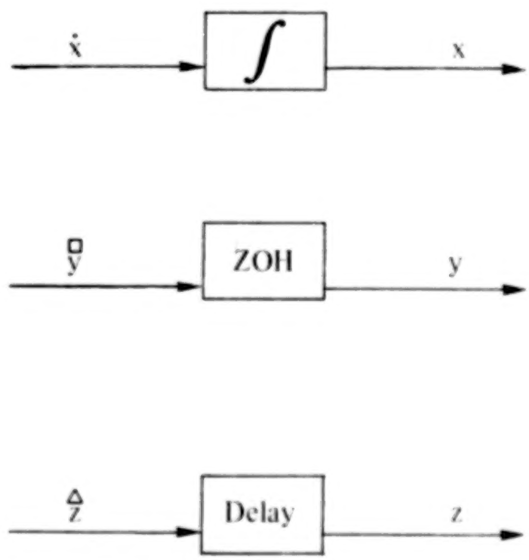
The concepts presented here stem largely from ideas derived from the work of Kalman and Bertram, 1959 (Reference 4), Zadeh and Desoer, 1963 (Reference 11), and National Aeronautics and Space Administration/Goddard Space Flight Center (NASA/GSFC) Solar Maximum Mission (SMM) project support activities, Donohue et al, 1978 (Reference 2).

Consider the problem of developing a simulation model for a linear continuous plant controlled by a linear sampled data control system. If this is done on an analog computer with digital components, then the plant equations are implemented by a set of summers, amplifiers and integrators, while the controller equations are implemented by a set of zero order holds, digital delays and some more summers, amplifiers and integrators. If this is to be done on a digital computer, the problem is not as straightforward due to the fact that zero order holds and digital delayors cannot be characterized by linear ordinary differential equations. Furthermore, if one blindly proceeds with conventional numerical integration methods, the net outcome is frequently a computationally costly solution.

To arrive at a mathematical formulation compatible with digital simulation methods, we first look through the eyes of a mathematician at the bag of hardware required for the analog simulation. Summers and amplifiers are memoryless objects, while integrators, zero order holds and delayors are

finite memory objects. Carrying this a step further, we can view the integrator as a piece of hardware which implements the mathematical rules of integration; similarly, it is just as valid to view the piece of hardware called a zero order hold as the implementation of the mathematical rules of zero order hold and to view the piece of hardware called a digital delayor as the implementation of the mathematical rules of digital delay.

At this point, drop the notion of hardware and recognize that new mathematical operations have been defined, i.e., the mathematical operations of zero order hold and that of digital delay. These readily lend themselves to block diagram representation and can be pictorially represented as

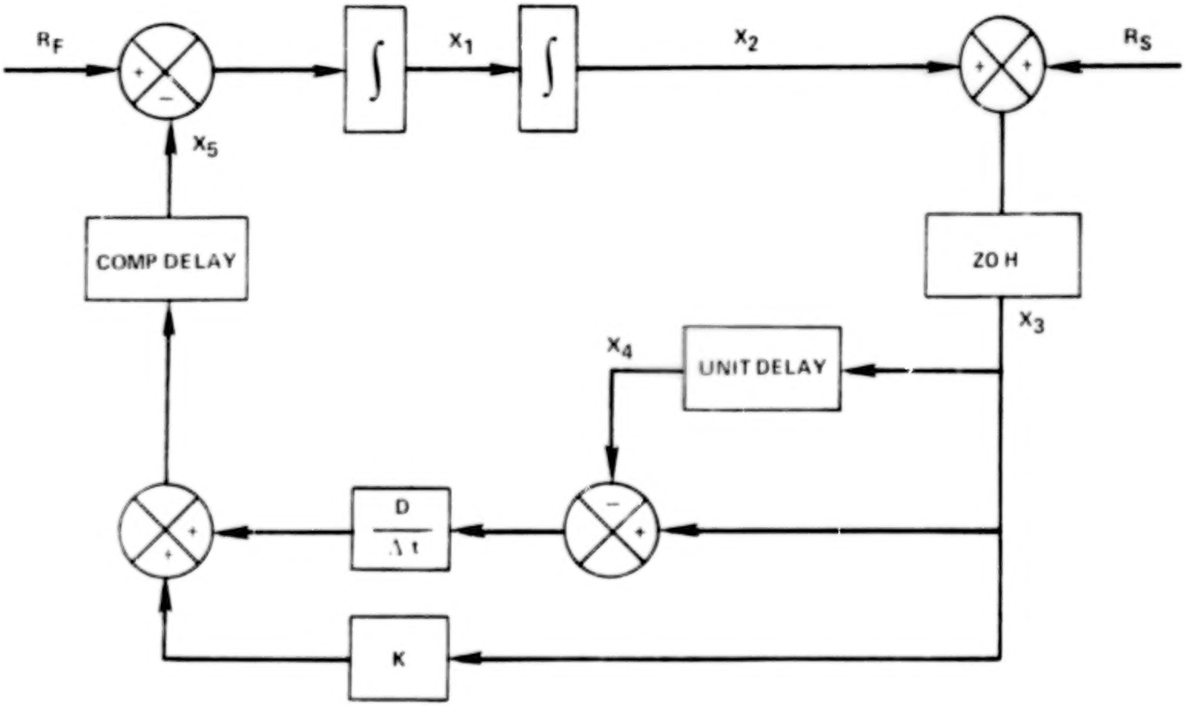


where (·) dot calls for the mathematical operation of integration, (□) box calls for zero order hold and (Δ) triangle calls for digital delay.

If several zero order holds having different sampling rates and digital delayors having different delay times are required, the notation immediately is expandable by introducing new symbols such as (○) circle, (⬡) hexagon, etc., or more simply by a numbering sequence within the symbols.

How this ties in with an ability to define sampled data controllers in terms of mixed operation equations may not yet be completely obvious. Consider, therefore, the following problem: Define the

mixed operation equations for the single axis control of a unit mass via sampled position and computed rate feedback through a computational delay not equal to the sampling period. The block diagram representation follows:



In this figure the following notation is used; X_1 = plant rate, X_2 = plant position, X_3 = sampled position, $(X_3 - X_4)/\Delta t$ = computed rate, Δt = sampling period, K = position gain, D = rate gain, X_5 = control force, R_F = external force input, R_S = external position input. It immediately follows that the linear mixed operation equations which define this problem are:

$$\begin{Bmatrix} \dot{X}_1 \\ \dot{X}_2 \\ \overset{\square}{X}_3 \\ \overset{\triangle}{X}_4 \\ \overset{\hexagon}{X}_5 \end{Bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \left(K + \frac{D}{\Delta t}\right) & -\frac{D}{\Delta t} & 0 \end{bmatrix} \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{Bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} R_F \\ R_S \end{Bmatrix} \qquad (1)$$

The important point to note in this example is the ease with which the linear constant coefficient mixed operation equations have been defined directly from the block diagram. The constant coefficient matrix, along with the rules associated with the mathematical operations used, completely defines the problem and can be readily implemented in a format compatible with digital computation.

The objective now is to define the steps required to go from linear mixed operation equations to linear finite difference equations. Once these are in hand, follow-on frequency domain studies can be carried out. If linear time domain response is required, the finite difference equations can be coded and time response obtained without recourse to conventional numerical integration methods. If nonlinear time domain response is required, a framework for developing special purpose integration algorithms which account for the digital signal characteristics of certain state variables is provided. In application, these numerical algorithms must often be developed to achieve timewise efficient simulation programs.

In the next section, the concept of mixed operation equations is presented in a more generalized framework and certain limitations imposed by the demand that the end product be computationally efficient are discussed.

GENERALIZED FORMULATION, ASSUMPTIONS AND LIMITATIONS

Kalman and Bertram (Reference 4) define several different types of sampling systems, e.g., synchronous, nonsynchronous, multirate, random, etc. There is no intention here to cover all possible cases. This formulation assumes that a single clock, contained within the system, controls all sampling. The clock is assumed to provide a never-ending sequence of synchronous clock pulses, the period between each clock pulse being exactly Δt. All sampling is carried out by the hardware items heretofore referred to as zero order holds. Each zero order hold is assumed to be synchronous and to have its sampling period defined as an integer multiple of the clock pulse period Δt. Zero order holds in any simulation may have different sampling periods; however, if frequency domain analysis is the goal, there must be a periodic pattern to the sampling. If not, frequency domain analysis has no meaning.

In application, the penalty on generality paid for by restricting the sampling periods to be integer multiples of Δt is minimal. This is adequate for most linear and quasi-linear system design and performance evaluation studies, and it leads to mathematical requirements that can be implemented so as to yield timewise efficient simulation models.

The need to incorporate the effects of one or more delays in a reasonably general format presents problems, if one demands that the end product be useful for general application. Delays can be placed into three general categories: 1) digital delay with time constant equal to an integer multiple of Δt, 2) digital delay with time constant not equal to an integer multiple of Δt, and 3) analog or transport delays.

Analog delays are not considered here for the simple reason that the author has not been able to find a method which he deems to be computationally practical for timewise efficient simulations.

Digital delays are considered and are implemented by the use of what will be referred to here as unit delayors and fractional delayors subject to the following assumptions and limitations: 1) a unit delayor delays a digital signal, which is constant during the clock pulse period Δt, by the amount Δt, and 2) a fractional delayor delays a digital signal, which is constant during the clock pulse period Δt, by the amount $\gamma \Delta t$ where $0 < \gamma < 1$. These limitations are incorporated to prohibit the modeling of a delayed analog signal by a series connection of several fractional delayors, and to set up a framework which will lead to tractable computational requirements.

Consider a general multivariable sampled data control system subject to the above assumptions and limitations, and let:

W – column matrix containing state variables associated with all integrators

X – column matrix containing state variables associated with all zero order holds

Y – column matrix containing state variables associated with all unit delayors

Z – column matrix containing state variables associated with all fractional delayors

• – symbol for integration

$\square$ – symbol for zero order hold: subpartitioned into $\boxed{1}$, $\boxed{2}$, . . . etc., if many different zero order holds are used

$\triangle$ – symbol for unit delay

$\bigcirc$ – symbol for fractional delay, subpartitioned into ① , ② , . . . etc., if many different fractional delays are used

Based upon the above comments and notation, the general form for a system of linear mixed operation equations is

$$\begin{Bmatrix} \dot{W} \\ \overset{\square}{X} \\ \overset{\triangle}{Y} \\ \overset{\bigcirc}{Z} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ 0 & A_{32} & A_{33} & A_{34} \\ 0 & A_{42} & A_{43} & 0 \end{bmatrix} \begin{Bmatrix} W \\ X \\ Y \\ Z \end{Bmatrix} + \begin{bmatrix} B_1 \\ B_2 \\ 0 \\ 0 \end{bmatrix} \begin{Bmatrix} R \end{Bmatrix} \tag{2}$$

where

A,B – partitions with nonzero elements

0 – partitions with all zero elements

R – column matrix of reference inputs assumed to be continuous functions of time; these cannot feed directly into a delayor.

It is also possible to define systems of nonlinear mixed operation equations. These, in general, will not painlessly lead to finite difference equations. However, at times they do lead to special-purpose integration algorithms which are superior to those specifically developed for systems of first order nonlinear ordinary differential equations. The user is cautioned that relative superiority must be judged on a problem-by-problem basis and is strongly dependent upon the talents of the analyst/programmer.

MIXED MATHEMATICAL OPERATIONS

The system of linear mixed operation equations as presented is relatively easy to obtain directly from a block diagram definition of a controller. However, in order for these to be useful for follow-on analysis, they must lead to a set of either linear ordinary differential or linear finite difference equations in an automatable manner. The objective of this section is to outline the methodology to be used to automate the procedure by which system state at clock time $(k+1)\Delta t$ can be expressed as a linear function of system state and input at clock time $k\Delta t$, $k = 0,1,2,....$ This is the desired set of first order linear finite difference equations which, via use of the z-transform, will lead directly into follow-on frequency domain analysis methods.

Let $W(k)$, $X(k)$, $Y(k)$, $Z(k)$ define system state and let $R(k)$ define system input at clock time $k\Delta t$. To define system state at time $(k+1)\Delta t$, it is necessary to know the time dependency of all system state variables and inputs during the Δt time interval

$$k\Delta t \leq t \leq (k+1)\,\Delta t$$

This is known via the modeling restrictions imposed, and from the mathematical rules associated with the mixed operation equations given below as:

- Integration: The matrix equation which defines all integrator state variables takes on the general form

$$\dot{W}(t) = A_{11}W(t) + U(t) \tag{3}$$

where U(t) is everything in row 1 of equation 2 not explicitly stated herein. Let W(k) define a full set of initial conditions for equation 3. It follows from the mathematical rules associated with the operation of integration that the state of all integrator state variables at any time t in the time interval

$$k\Delta t < t \leqslant (k+1)\Delta t$$

is given by

$$W(t) = e^{A_{11}(t - k\Delta t)}W(k) + \int_{k\Delta t}^{t} e^{A_{11}(t-\tau)}\, U(\tau)\, d\tau \tag{4}$$

where it is immediately pointed out that numerical algorithms for evaluating the matrix exponential, and an integral involving the matrix exponential, will be required. From the standpoint of computational mathematics, the numerics associated with the evaluation of equation 4 will be practical only if the components of $U(\tau)$ are digital and delayed digital signals with no more than one jump discontinuity in the integration interval. This is the reason for the modeling restrictions which prohibit analog delayors and series connections of fractional delayors.

- Zero Order Hold: The matrix equation which defines all zero order hold state variables takes on the general form

$$\overset{\square}{X}(t) = U(t) \tag{5}$$

where U(t) now is everything on the right-hand side of equation 2 in row 2. A zero order hold samples the incoming signal at the sampling instant, it updates its output signal an instant later and then holds that constant value until it is subsequently updated an instant after the next sampling instant. The output signal of a zero order hold is frequently referred to herein as a digital signal. The above descriptive definition of what a zero order hold does is encapsulated in the mathematical solution to the zero order hold equation (5) given by

$$X(t) = U(k)$$

for

$$k\Delta t < t \leqslant (k+1)\Delta t \tag{6}$$

If zero order holds with different sampling rates must be studied, one cannot blindly apply equation 6. For example, assume a zero order hold with sample period $n\Delta t$. A trivial notation extension allows one to symbolically express its equation as

$$\overset{\boxed{n}}{X}(t) = U(t) \tag{7}$$

If clock time $k\Delta t$ is a sampling instant for this zero order hold, then the solution to equation 7 becomes

$$X(t) = U(k) \tag{8}$$

for

$$k\Delta t < t \leqslant (k+n)\Delta t$$

- Unit Delay: The matrix equation which defines all unit delay state variables takes on the general form

$$\overset{\triangle}{Y}(t) = U(t) \tag{9}$$

where $U(t)$ now is everything on the right-hand side of equation 2 in row 3. A unit delay reads the incoming digital signal at the sampling instant $k\Delta t$, and then outputs the state it was at, at $k\Delta t$, until an instant before the next sampling instant at which time it updates its output signal. Mathematically, this is encapsulated in the solution to the unit delay equation which is given by

$$Y(t) = \begin{cases} Y(k) & \text{for } k\Delta t < t < (k+1)\Delta t \\ U(k) & \text{for } t=(k+1)\Delta t \end{cases} \tag{10}$$

- Fractional Delay: The matrix equation which defines all fractional delay state variables takes on the general form

$$\overset{\bigcirc}{Z}(t) = U(t) \tag{11}$$

where $U(t)$ now is everything on the right-hand side of equation 2 in row 4. A fractional delay reads the incoming digital signal at the sampling instant $k\Delta t$. It then outputs the value of its state, at $k\Delta t$, until the time $(k+\gamma)\Delta t$ at which time it updates its output signal. Its output signal is referred to

herein as a delayed digital signal. Mathematically, this is encapsulated in the solution to the fractional delay equation, i.e.,

$$Z(t) = \begin{cases} Z(k) & k\Delta t < t \leqslant (k+\gamma)\Delta t \\ U(k) & (k+\gamma)\Delta t < t \leqslant (k+1)\Delta t \end{cases} \tag{12}$$

where $0 < \gamma < 1$. If fractional delayors with different delay times are required for simulation, subscripts would be required on the γ's and in the symbol ○ in a manner similar to that used in the zero order hold discussion.

SOLUTION OF MIXED OPERATION EQUATIONS

The most general form for linear mixed operation equations considered in the context of this paper is presented in equation 2. This can be expressed as a set of first order finite difference equations by direct application of the closed-form solutions for each operation considered in the previous paragraphs. In keeping with the author's intent to communicate to the reader, the solution presented herein assumes that all zero order holds have sampling period Δt and all fractional delays have delay time $\gamma\Delta t$. The solution for the multizero order hold, multifractional delay situation requires additional notation which on the bottom line is more confusing than illuminating; it will therefore not be presented here.

The row 1 equation of equation 2 defines the integrator state variables and is given by

$$\dot{W}(t) = A_{11}\, W(t) + A_{12}\, X(t) + A_{13}\, Y(t) + A_{14}\, Z(t) + B_1\, R(t) \tag{13}$$

In order to concisely define the solution to this equation let

$$F(\Delta t) = e^{A_{11}\Delta t} \tag{14}$$

$$H(\Delta t) = \int_0^{\Delta t} e^{A_{11}(\Delta t-\tau)}\, d\tau \tag{15}$$

$$H[(1-\gamma)\Delta t] = \int_{\gamma\Delta t}^{\Delta t} e^{A_{11}(\Delta t-\tau)}\, d\tau \tag{16}$$

and note that

$$H(\gamma\Delta t) = H(\Delta t) - H[(1-\gamma)\Delta t] \tag{17}$$

Direct application of equation 4 evaluated at $t = (k+1)\Delta t$ along with equations 6, 8, 10 and 12 leads to the finite difference equation

$$\begin{aligned} W(k+1) = {} & F(\Delta t)\, W(k) + A_{12}\, H(\Delta t)\, X(k+1) \\ & + A_{13}\, H(\Delta t)\, Y(k) \\ & + A_{14} \left\{ H(\gamma\Delta t)\, Z(k) + H[(1-\gamma)\Delta t]\, Z(k+1) \right\} \\ & + B_{1}\, H(\Delta t)\, R(k) \end{aligned} \tag{18}$$

where the assumption is made that over the Δt time period the input $R(t)$ is constant and equal to $R(k)$. Furthermore, note the use of $X(k+1)$ and $Z(k+1)$ in the right-hand side of equation 18. This is a statement of the fact that over the interval of integration, these state variables are defined by their state at $(k+1)\Delta t$.

The row 2 equation of equation 2 defines the zero order hold state variables. Direct application of equation 6 leads to the finite difference equation

$$X(k+1) = A_{21} W(k) + A_{22} X(k) + A_{23} Y(k) + A_{24} Z(k) + B_{2}\, R(k) \tag{19}$$

The row 3 equation of equation 2 defines the unit delay state variables. Direct application of equation 10 leads to the finite difference equation

$$Y(k+1) = A_{32} X(k) + A_{33} Y(k) + A_{34}\, Z(k) \tag{20}$$

The row 4 equation of equation 2 defines the fractional delay state variables. Direct application of equation 12 leads to the finite difference equation

$$Z(k+1) = A_{42}X(k+1) + A_{43}Y(k+1) \tag{21}$$

Equations 18, 19, 20 and 21 can now be combined to yield the following matrix of simultaneous first order finite difference equations

$$\begin{bmatrix} 1 & -A_{12}H(\Delta t) & 0 & -A_{14}H[(1-\gamma)\Delta t] \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -A_{42} & -A_{43} & 1 \end{bmatrix} \begin{Bmatrix} W(k+1) \\ X(k+1) \\ Y(k+1) \\ Z(k+1) \end{Bmatrix}$$

$$= \begin{bmatrix} F(\Delta t) & 0 & A_{13}H(\Delta t) & A_{14}H(\gamma\Delta t) \\ A_{21} & A_{22} & A_{23} & A_{24} \\ 0 & A_{32} & A_{33} & A_{34} \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} W(k) \\ X(k) \\ Y(k) \\ Z(k) \end{Bmatrix} + \begin{bmatrix} B_1 H(\Delta t) \\ B_2 \\ 0 \\ 0 \end{bmatrix} \{R(k)\} \tag{22}$$

Direct use of the inverse of the coefficient matrix on the left-hand side of the equation, which is

$$\begin{bmatrix} 1 & \{A_{12}H(\Delta t) + A_{14}H[(1-\gamma)\Delta t]A_{42}\} & \{A_{14}H[(1-\gamma)\Delta t]A_{43}\} & \{A_{14}H[(1-\gamma)\Delta t]\} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & A_{42} & A_{43} & 1 \end{bmatrix} \tag{23}$$

yields the final set of finite difference equations which are in a format compatible with follow-on analysis methods, i.e.,

$$\begin{Bmatrix} W(k+1) \\ X(k+1) \\ Y(k+1) \\ Z(k+1) \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{13} & D_{14} \\ D_{21} & D_{22} & D_{23} & D_{24} \\ D_{31} & D_{32} & D_{33} & D_{34} \\ D_{41} & D_{42} & D_{43} & D_{44} \end{bmatrix} \begin{Bmatrix} W(k) \\ X(k) \\ Y(k) \\ Z(k) \end{Bmatrix} + \begin{bmatrix} E_1 \\ E_2 \\ E_3 \\ E_4 \end{bmatrix} \{R(k)\} \tag{24}$$

DIGITAL COMPUTATIONAL REQUIREMENTS

Wilkinson (Reference 10) reminds us that "attractive mathematics does not protect one from the rigors of digital computation." It is therefore important to determine whether or not the foregoing presentation has led to a computationally practical methodology for the particular case worked out and for the multizero order hold, multifractional delay case mentioned.

The following sequence of steps is required to go from problem definition to the desired set of finite difference equations to be used for either time or frequency domain follow-on analysis.

Step 1. Set up the linear mixed operation equations in the partitioned matrix format of equation 2. This is straightforward and can be done by hand or problem-unique computer code.

Step 2. Determine the elements in the coefficient matrices on the right- and left-hand side of equation 22. The major numerical problem here is in the evaluation of the matrix exponential $F(\Delta t)$ and the integrals involving the matrix exponential, namely, $H(\Delta t)$ and $H(\gamma\Delta t)$. These matrices can be efficiently and accurately evaluated by use of subroutine PADE developed by Van Loan and discussed in Reference 8, discussed and published in Reference 9.

Step 3. Determine the elements in the coefficient matrices of equation 24. These are obtainable by a simple matrix multiply, since the required matrix inverse is definable in closed form.

FREQUENCY DOMAIN ANALYSIS

If frequency domain analysis is the objective, a periodic pattern to the sampling must exist. If there exists a periodic pattern to the sampling, then it is possible via successive matrix multiplications to arrive at an equation of the form

$$\begin{Bmatrix} W(k+N) \\ X(k+N) \\ Y(k+N) \\ Z(k+N) \end{Bmatrix} = \begin{bmatrix} \psi_{11} & \psi_{12} & \psi_{13} & \psi_{14} \\ \psi_{21} & \psi_{22} & \psi_{23} & \psi_{24} \\ \psi_{31} & \psi_{32} & \psi_{33} & \psi_{34} \\ \psi_{41} & \psi_{42} & \psi_{43} & \psi_{44} \end{bmatrix} \begin{Bmatrix} W(k) \\ X(k) \\ Y(k) \\ Z(k) \end{Bmatrix} + \begin{bmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \\ \nu_4 \end{bmatrix} \left\{ R(k) \right\} \tag{25}$$

where $N\Delta t$ is the sampling pattern period and the elements of the coefficient matrix are constant and independent of both k and N. Kalman and Bertram (Reference 4) discuss this and refer to this matrix as a stationary transition matrix. It should be noted that in equation 25 the standard stability analysis assumption that input R(k) is constant over the pattern period $N\Delta t$ has been made; if not made, the resultant stationary finite difference equation is unmanageable for follow-on frequency domain analysis.

In order to transform time domain equation 25 into a frequency domain equation of standard form, let $\Delta T = n\Delta t$ define the sampling pattern period and define the z transform of the arbitrary discrete time function $f(n\Delta T)$ n=o, 1, 2, ...as

$$f(z) = \sum_{n=0}^{\infty} f(n\Delta T)\, z^{-n} \tag{26}$$

Recognize that $W(n\Delta T)$, $X(n\Delta T)$,... and $R(n\Delta T)$ are discrete time functions, utilize equations 26 to transform equation 25 and obtain the desired set of standard form frequency domain equations, i.e.,

$$[z\mathbb{1} - \psi] \begin{Bmatrix} W(z) \\ X(z) \\ Y(z) \\ Z(z) \end{Bmatrix} = [\nu] \quad R(z) \tag{27}$$

where $\mathbb{1}$ is the unit diagonal matrix and the matrices ψ and ν are the coefficient matrices defined in equation 25. Procedural details required for the above transformation can be found in Reference 3 (Jury, 1964).

The reader should note the author's intentional avoidance of the mention of the Laplace transform variable s in the z-transform definition used. From the standpoint of computational mathematics, it is the author's opinion that numerical computation should be carried out with the z-domain equations provided in equation 27. Once z-domain results are computationally obtained, they can always be mapped into:

The s-domain by the mapping equation

$$z = e^{s\Delta t}$$

The w-domain by the mapping equation

$$z = \frac{1 + w}{1 - w}$$

or the ζ-domain by the mapping equation

$$\zeta = \frac{z - 1}{\Delta t}$$

This final mapping step should be viewed as a necessity since results must be presented in a format compatible with the desires of control design engineers. This computational approach avoids truncated series approximations, and the resultant numerical computation pitfalls which can be encountered, when computation is done with s-domain equations.

Determination of closed loop system stability can be made directly from the eigenvalues of the coefficient matrix in equation 25; these may be computed via application of the EISPACK library of matrix eigensystem routines published in Reference 7 (Smith, et al., 1976). If all eigenvalues are contained within the unit circle in the complex z-domain, then the system is stable. If any eigenvalue is outside of the unit circle, the system is unstable.

Determination of margins of stability is usually accomplished with opened, partially opened and closed loop z-domain transfer functions. A methodology for generating any of these type transfer functions which is based upon a state variable formulation of system response is presented in Reference 1 (Bodley, et al., 1978) and implemented in the DISCOS program defined therein. Even though the DISCOS method is specifically for continuous time systems, and hence s-domain transfer functions, slight modifications can be introduced which make it applicable for z-domain transfer functions.

The DISCOS method opens control feedback loops by selectively setting elements to zero in the coefficient matrix of the set of first order ordinary differential equations used to define system response. This resultant coefficient matrix is the coefficient matrix for an opened loop system, and resultant closed loop transfer functions are in actuality the sought-after open loop transfer functions of the original unabridged system. To apply the method to a sampled data system, it is important to recognize that feedback loops must be opened by setting to zero elements in the coefficient matrix of the mixed operation equations. One then proceeds directly as defined herein to obtain the coefficient matrix for the finite difference equations, which define open loop system response, and then on to z-domain transfer function analysis.

TIME DOMAIN ANALYSIS

If time domain analysis is the objective, the concept of mixed operation equations, at times, can lead to tailor-made integration methods which, in particular applications, can reduce computation run time by orders of magnitude.

For example, if the matrix A_{11} in equation 2 is constant and input **R** is constant over each Δt sampling interval, classic numerical integration methods for the differential equations can be completely avoided. One needs only pass through the matrix exponentiation routine PADE once, save the results, and then compute time domain response directly from the finite difference equations provided by equation 24. If the matrix A_{11} is slowly varying relative to Δt, it often will be found to be more efficient to call up the matrix exponentiation routine periodically to update data for equation 24, rather than blindly use classical numerical integration methods.

The reader should recognize that numerical integration methods for differential equations are the subject of active research; however, most of the current work being done is concentrated on what in the literature is referred to as stiff ordinary differential equations (S' ›mpine, et al., 1976 and 1979, References 5 and 6, and Gear, 1981, Reference 12). Methods for integrating equations of the type encountered in the sampled data control of oscillatory plants, as defined herein, is an area of research not being addressed in the open literature today. Hopefully, the ideas presented in this paper will stimulate research into the field of numerical integration of mixed operation equations. This is an important class of equations for which timewise efficient methods for numerical solution are now developed on a problem-by-problem basis.

CONCLUSION

The methodology presented here is a generalization of methods successfully used at NASA/GSFC during a control system stability and performance analysis of several modes of operation for the SMM spacecraft. In that application equation 24 was transformed by use of the z-transform, so that stability margins could be determined in the frequency domain. As a benchmark; in that application, 20 different modes of spacecraft operation were analyzed, each requiring the regeneration of the coefficient matrix in equation 24. Seventeen state variables were required to define the plant and controller, and all four mathematical operations discussed herein were utilized. Upper and lower gain margins were obtained in a straightforward iterative manner by incrementing feedback gain and checking for the gain values which put system roots (eigenvalues) on the threshold of instability, i.e., out of the z-domain unit circle. Once debugged, the entire job was run in 1.6 minutes CPU on the IBM 360-91. Results were compared with those obtained by Donohue, et al., 1978, (Reference 2), using more conventional s-domain methods. Agreement in all cases was near perfect.

Donohue, Hager, McGlew and Zimmerman also carried out time domain performance studies for the SMM spacecraft. The tailor-made integration method used in their study contributed significantly to the ideas used and presented here in a generalized context. Their experience conclusively demonstrated that for certain real application problems, dramatic (orders of magnitude) improvements in computer run-time could be obtained if one avoided the blind application of conventional numerical integration methods for sampled data control equations.

PROGRAM ACRONYM/AVAILABILITY LIST

ACSL	Advanced Continuous Simulation Language; Mitchell & Gauthier Associates, Inc., Concord, Massachusetts
CSMP	System/360 Continuous System Modelling Program, IBM Program Number 360A-CX-16X
CSSL	Continuous System Simulation Language; Simulation Services, Division of Nilsen Associates, Inc., Chatsworth, California
DARE	Differential Analyzer Replacement; University of Arizona
DISCOS	Digital Computer Program for the Dynamic Interaction Simulation of Controls and Structure, COSMIC Program Number GSC-12422
MODEL	Multi-Optimal Differential Equation Language; NASA/GSFC, Guidance & Control Branch Program
NASTRAN	NASA Structural Analysis; MacNeil Schwendler Corp., Los Angeles, California, and/or COSMIC, University of Georgia, Athens, Georgia, Program Number HQN-10952
SINDA	Systems Improved Numerical Differencing Analyzer, COSMIC Program Number MSC-13805
SPAR	A System of Computer Programs for Structural Analysis, COSMIC Program Number LAR-12213; also under title EAL from Engineering Information Systems, Inc., San Jose, California.
TRASYS	Thermal Radiation Analysis System, Martin-Marietta and/or NASA/JSC Computer Program Library

REFERENCES

1) Bodley, C.S., Devers, A.D., Park A.C. and Frisch, H.P., "A Digital Computer Program for the Dynamic Interaction Simulation of Controls and Structure (DISCOS)," NASA Technical Paper 1219, Vols. 1 and 2, May 1978.

2) Donohue, J.H., Hager, F.W., McGlew, D.E. and Zimmerman, B.G., NASA/GSFC Guidance and Control Branch Internal Memoranda, 1978.

3) Jury, E.I., *Theory and Application of the Z-Transform Method*, Ch. 1, John Wiley and Sons, 1964.

4) Kalman, R.E. and Bertram, J.E., "A Unified Approach to the Theory of Sampling Systems," *Journal of the Franklin Institute*, Vol. 267, pp. 405-436, May 1959.

5) Shampine, L.F. and Gear, G.W., "A User's View of Solving Stiff Ordinary Differential Equations," *SIAM Review*, Vol. 21, No. 1, pp. 1-17, January 1979.

6) Shampine, L.F., Watts, H.A., Davenport, S.M., "Solving Nonstiff Ordinary Differential Equations – The State of the Art," *SIAM Review*, Vol. 18, No. 3, pp. 376-411, July 1976.

7) Smith, B.T., Boyle, J.M., Dongarra, J.J., Garbow, B.S., Ikebe, Y., Klema, V.C. and Moler, C.B., "Matrix Eigensystem Routines – EISPACK Guide," Lecture Notes in *Computer Science 6*, Springer-Verlag, 1976.

8) Van Loan, C.F., "Computing Integrals Involving the Matrix Exponential," *IEEE Trans. Automatic Control*, Vol. AC-23, No. 3, June 1978 (no subroutine source code).

9) Van Loan, C.F., "Computing Integrals Involving the Matrix Exponential," Dept. of Computer Science, Cornell University, Ithaca, New York, TR76-298 (with subroutine source code).

10) Wilkinson, J.H., "Modern Error Analysis," *SIAM Review*, Vol. 13, No. 4, pp. 548-568, October 1971.

11) Zadeh, L.A. and Desoer, C.A., *Linear System Theory, The State Space Approach*, McGraw-Hill, 1963.

12) Gear, C.W., "Numerical Solution of Ordinary Differential Equations: Is there anything left to do?" *SIAM Review*, Vol. 23, No. 1, pp. 10-24, January 1981.

1. Report No. NASA TP-1817	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Time and Frequency Domain Analysis of Sampled Data Controllers via Mixed Operation Equations		5. Report Date June 1981
		6. Performing Organization Code 712
7. Author(s) Harold P. Frisch		8. Performing Organization Report No. 81F0042
9. Performing Organization Name and Address Goddard Space Flight Center Greenbelt, Maryland 20771		10. Work Unit No.
		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Paper
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		
		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

Specification of the mathematical equations required to define the dynamic response of a linear continuous plant, subject to sampled data control, is complicated by the fact that the digital components of the control system cannot be modeled via linear ordinary differential equations. This complication can be overcome by introducing two new mathematical operations; namely, the operation of zero order hold and digital delay. It is shown herein that by direct utilization of these operations, a set of linear mixed operation equations can be written and used to define the dynamic response characteristics of the controlled system. It also is shown how these linear mixed operation equations lead, in an automatable manner, directly to a set of finite difference equations which are in a format compatible with follow-on time and frequency domain analysis methods.

17. Key Words (Selected by Author(s)) Analytical and Numerical Methods, Guidance and Control, Spacecraft Simulation		18. Distribution Statement Unclassified-Unlimited Subject Category 18	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 25	22. Price* A02

*For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1981

20

90%

END

10-16-81